IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellants:	Joseph G. Laura, et al.	§	
		§	Group Art Unit: 2453
Serial No.:	10/682,463	§	
		§	Examiner: Higa, Brendan Y.
Filed:	October 9, 2003	§	
		§	Confirmation No.: 9210
For:	SYSTEM FOR PROCESSING DATA RETRIEVED FROM AN INFORMATION SERVICE LAYER	§ § §	

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Dear Sirs:

This Appeal Brief is filed in support for the appeal in the above referenced application and is filed pursuant to the Notice of Appeal filed November 23, 2009 and the Notice of Panel Decision from Pre-Appeal Brief Review dated December 18, 2009. Appellants authorize all required fees under 37 C.F.R. § 1.17 to be charged to Deposit Account No. 21-0765, of Sprint Communications Company L.P.

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST 4

II. RELATED APPEALS AND INTERFERENCES 5

III. STATUS OF CLAIMS 6

IV. STATUS OF AMENDMENTS 7

V. SUMMARY OF CLAIMED SUBJECT MATTER 8

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL 17

VII. ARGUMENT 18

A. 35 U.S.C. § 103 Rejections - Claims 1, 6-8, 11, 16, 18, and 20-30. 20

1. Whether claims 1, 6-8, 11, 16, 18, and 20-30 are unpatentable under 35 U.S.C. § 103(a) as being obvious over Draper in view of Brown. 20

a. Claims 1, 6-8, 11, 16, 18, and 20-30 were wrongly rejected because neither Draper nor Brown, alone or in combination, disclose an analyzer that analyzes the at least a portion of the requested data wherein the analyzer performs calculations on the at least a portion of the requested data according to the at least one information command layer script received from the application program interface. 20

b. Claims 1, 6-8, 11, 16, 18, and 20-30 were wrongly rejected because neither Draper nor Brown disclose formatting the requested data twice. 23

2. Whether claims 4, 5, and 19 are unpatentable under 35 USC § 103(a) as being obvious over Draper in view of Brown and in further view of Zarb. 27

a. Claims 4 and 5 depend from claim 1 and were wrongly rejected for at least the reasons established in sections VII.A.1.a. and VII.A.1.b. 27

b. Claim 19 depends from claim 16 and was wrongly rejected for at least the reasons established above in sections VII.A.1.a. and VII.A.1.b. 27

VIII. CONCLUSION 28

IX. CLAIMS APPENDIX 29
X. EVIDENCE APPENDIX 37

XI. RELATED PROCEEDINGS APPENDIX 38

I. REAL PARTY IN INTEREST

The real party in interest in the present application is the following party: Sprint Communications Company L.P.

II. RELATED APPEALS AND INTERFERENCES

None.

III. STATUS OF CLAIMS

A. Total Number of Claims in the Application

The claims in the application are: 1, 4-8, 11, 16, and 18-30

B. Status of All Claims in the Application

1. Claims canceled: 2-3, 9-10, 12-15, and 17
2. Claims withdrawn from consideration but not canceled: None
3. Claims pending: 1, 4-8, 11, 16, and 18-30
4. Claims allowed: None
5. Claims rejected: 1, 4-8, 11, 16, and 18-30

C. Claims on Appeal

The claims on appeal are: 1, 4-8, 11, 16, and 18-30

IV. STATUS OF AMENDMENTS

No amendments were filed after the August 27, 2009 Final Office Action (hereinafter "Final Office Action").

V. SUMMARY OF CLAIMED SUBJECT MATTER

It is vital in a large corporate enterprise to have management and operations information available in a timely and cost effective manner. *See, e.g.,* Application at 1, ¶ [0005], line 7-Application at 2, ¶ [0005], lines 1-2.[1] The information may be useful in calculating costs as a function of the level of performance rather than just the cost of operating. *See, e.g.,* Application at 1, ¶ [0005], lines 6-7. These and other calculations may be valuable to managers and executives making business decisions for the enterprise. *See, e.g.,* Application at 1, ¶ [0005], line 7-Application at 2, ¶ [0005], lines 1-2. However, in a large corporate enterprise, management and operations information may be spread across multiple unrelated computer systems. *See, e.g.,* Application at 1, ¶ [0005], lines 1-2. If the multiple unrelated computer systems are incompatible with one another, the management and operations information may not be available, may not be available in a timely fashion, or may not be available in a cost effective manner. *See, e.g.,* Application at 2, ¶ [0005], lines 2-4.

Thus, the pending application discloses systems and methods that provide translation/transformation of data, such as messaging data, which is collected by a distributed management system, for collection, consolidation, processing, and analysis. *See, e.g.,* Application at 1, ¶ [0004], lines 1-3; Application at 25, ¶ [0069], lines 4-5. This data may be stored in a plurality of databases and may be accessed by a service layer by using an information service layer (ISL) script. *See, e.g.,* Application at 2, ¶ [0006], lines 5-6; Application at 25, ¶

---

[1] 37 C.F.R. § 41.37 (c)(1)(v) provides that the "[s]ummary of claimed subject matter . . . shall refer to the specification by page and line number." The instant application was presented in numbered page and numbered paragraph form. As such, the citations to the specification support for the claimed subject matter will be presented in the following form: Application at ___ (page number), ¶ [____] (paragraph number), lines ____ (lines within the corresponding paragraph). On the occasion when the pertinent paragraph is contained on multiple pages, the paragraph line numbering will begin anew on subsequent pages.

[0066], lines 2-7. The ISL script may search through diverse databases, extract desired data, and format the desired data in a usable format. *See, e.g.,* Application at 25, ¶ [0066], lines 1-5; Application at 25, ¶ [0067], lines 1-3. For example, the ISL script may wrap the extracted data in tags, such as a document type definition (DTD) tag. *See, e.g.,* Application at 25, ¶ [0066], lines 5-8; Application at 25, ¶ [0067], lines 3-4; Application at 12, ¶ [0045], lines 8-11-Application at 13, ¶ [0045], lines 1-2. This allows data that may come from diverse sources to be understood and used for analysis together. *See, e.g.,* Application at 6, ¶ [0025], lines 1-4; Application at 25, ¶ [0067], lines 1-3.

An information command layer (ICL) script comprises embedded instructions or logic that may be used to manipulate the data that has been retrieved from the one or more databases and formatted by the ISL script. *See, e.g.,* Application at 25, ¶ [0069], lines 1-2; Application at 26, ¶ [0070], lines 1-3. This data manipulation may, in some embodiments, provide instructions to parse, add, subtract, or manipulate the data from one or more diverse databases. *See, e.g.,* Application at 26, ¶ [0071], lines 1-2. The ICL script may be provided as an input parameter through the API by an application or be stored locally on the ICL. *See, e.g.,* Application at 26, ¶ [0070], lines 4-8. Since the ICL script is not hard coded, different ICL scripts can be used to easily perform different types of analysis. *See, e.g.,* Application at 26, ¶ [0070], lines 8-9. Moreover, a writer component is also disclosed that further formats the analyzed data into a format appropriate for the application that requested the data. S*ee, e.g.,* Application at 25, ¶ [0069], lines 2-5.

Claim 1 is independent and is directed to a system for processing data stored on a computer readable medium retrieved from an information service layer, *see, e.g.*, Application at 25, ¶ [0067], line 1; Application at 25, ¶ [0069], lines 1-2, the system comprising: an application program interface that receives a request for data from an application, *see, e.g.*, Application at 25, ¶ [0069], lines 5-7, and that provides at least one information service layer script and at least one information command layer script *see, e.g.*, *id.*; Application at 26, ¶ [0070], lines 4-5; Application at 26, ¶ [0070], lines 9-11; a requestor component that builds a request for data using the at least one information service layer script, *see, e.g.*, Application at 25, ¶ [0066], lines 2-7, and transmits the built request for data to the information service layer, *see, e.g.*, Application at 25, ¶ [0066], lines 2-3; a receiver component communicating with the information service layer that receives a response from the information service layer, *see, e.g.*, Application at 25, ¶ [0067], line 1, the response formatted based upon the formatting described in the information service layer script, *see, e.g.*, Application at 25, ¶ [0067], lines 1-4, and the response including at least a portion of the requested data, *see, e.g.*, *id.*; an analyzer that receives the response from the receiver component, *see, e.g.*, Application at 25, ¶ [0069], lines 1-2, and the analyzer analyzes the at least a portion of the requested data, *see, e.g.*, Application at 26, ¶ [0071], lines 1-7, wherein the analyzer performs calculations on the at least a portion of the requested data according to the at least one information command layer script received from the application program interface, *see, e.g.*, Application at 25, ¶ [0069], lines 1-2; Application at 26, ¶ [0070], lines 1-5, Application at 26, ¶ [0071], lines 1-7, wherein instructions are embedded within the at least one information command layer script using a tag-based markup language, *see, e.g.*, Application at 26, ¶ [0070], line 3; Application at 13, ¶ [0045], lines 1-2, wherein the analyzer parses the information command layer script to obtain the instructions embedded in the information command layer script to analyze a metric associated with the application based on the at least a portion of the requested data, *see, e.g.*, Application at 18, ¶ [0062], line 10; Application at 20, ¶ [0062], line 7, and the analyzer creates an analyzed portion of the response, *see, e.g.*, Application at 26, ¶ [0071], lines 1-7; and a writer component formatting the analyzed portion of the response based upon a desired format, *see, e.g.*, Application at 25, ¶ [0069], lines 2-3; Application at 26, ¶ [0070], lines 1-3, wherein the application program interface sends the formatted analyzed portion of the response to the application, *see, e.g.*, Application at 25, ¶ [0069], lines 4-7.

Claim 4 depends on claim 1 and adds the limitation that the analyzer further verifies cycles associated with the application, *see, e.g.*, Application at 1, ¶ [0005], lines 2-6; Application at 31, ¶ [0087], lines 1-7.

Claim 5 depends on claim 4 and adds the limitation that the cycles associated with the application is further defined as invoicing cycles, *see, e.g.*, Application at 1, ¶ [0005], lines 2-6; Application at 31, ¶ [0087], lines 1-7.

Claim 6 depends on claim 1 and adds the limitation that the writer component formats the analyzed portion of the response as an extensible mark-up language report, *see, e.g.*, Application at 3, ¶ [0010], lines 7-8; Application at 24, ¶ [0064], lines 2-4; Application at 25, ¶ [0069], lines 2-3; Application at 26, ¶ [0070], lines 1-3; Application at 26, ¶ [0071], lines 6-7.

Claim 7 depends on claim 1 and adds the limitation that the writer component formats the analyzed portion of the response as an extensible mark-up language tag, *see, e.g.*, Application at 3, ¶ [0010], lines 7-8; Application at 12, ¶ [0045], lines 2-4, Application at 24, ¶ [0064], lines 2-4; Application at 25, ¶ [0069], lines 2-3; Application at 26, ¶ [0070], lines 1-3; Application at 26, ¶ [0071], lines 8-9.

Claim 8 depends on claim 1 and adds the limitation that the writer component formats the analyzed portion of the response as character delimited, *see, e.g.*, Application at 3, ¶ [0010], lines

7-8; Application at 24, ¶ [0064], lines 4-6; Application at 25, ¶ [0069], lines 2-3; Application at 26, ¶ [0070], lines 1-3.

Claim 11 depends on claim 1 and adds the limitation that the application program interface promotes communications between at least one application resident on at least one other computer system, *see, e.g.*, Application at 25, ¶ [0069], lines 5-7.

Claim 16 is independent and is directed to a method for processing data retrieved from an information service layer, *see, e.g.*, Application at 25, ¶ [0067], line 1; Application at 25, ¶ [0069], lines 1-2; the method comprising: building a request for data, *see, e.g.*, Application at 6, ¶ [0026], line 3-Application at 7, ¶ [0026], line 1, wherein the request comprises an information service layer script provided by an application program interface, *see, e.g.*, Application at 25, ¶ [0069], lines 5-7; Application at 26, ¶ [0070], lines 9-11; submitting the request for data to an information service layer, *see, e.g.*, Application at 10, ¶ [0038], line 1; Application at 25, ¶ [0066], lines 2-3; receiving at least a portion of the requested data from the information service layer, *see, e.g.*, Application at 25, ¶ [0067], line 1, wherein the at least a portion of the requested data is formatted according to the information service layer script, *see, e.g.*, Application at 25, ¶ [0067], lines 1-3; and processing the at least a portion of the requested data received from the information service layer according to an information command layer script *see, e.g.*, Application at 26, ¶ [0070], lines 1-3, wherein the information command layer script comprises tagged data, *see, e.g.*, Application at 12, ¶ [0045], lines 8-11; Application at 26, ¶ [0070], line 3, wherein the processing comprises: parsing the information command layer script, *see, e.g.*, Application at 18, ¶ [0062], line 10; Application at 20, ¶ [0062], line 7; interpreting embedded instructions from the information command layer script, *see, e.g.*, Application at 26, ¶ [0070], line 1-3; performing a calculation on the at least a portion of the requested data based upon the embedded instructions to analyze a metric associated with the application, *see, e.g.*, Application at 25, ¶ [0069], lines 1-2; Application at 26, ¶ [0070], lines 1-5, Application at 26, ¶ [0071], lines 1-7; and formatting the analyzed portion of the requested data based on a desired format, *see, e.g.*, Application at 25, ¶ [0069], lines 2-3; Application at 26, ¶ [0070], lines 1-3; wherein the information command layer script further comprises instructions to monitor a process based on the requested data and spawn an event based upon a trigger related to the requested data, *see, e.g.*, Application at 27, ¶ [0073], lines 2-3; Application at 27, ¶ [0074], lines 7-10; Application at 29, ¶ [0081], lines 4-9; Application at 29, ¶ [0082], lines 1-3.

Claim 18 depends on claim 16 and adds the limitation that the process is defined as an application operation, *see, e.g.*, Application at 1, ¶ [0005], lines 2-6; Application at 3, ¶ [0011], lines 1-2; Application at 7, ¶ [0030], lines 1-4.

Claim 19 depends on claim 16 and adds the limitation that the process is defined as an invoicing cycle, *see, e.g.*, Application at 1, ¶ [0005], lines 2-6; Application at 31, ¶ [0087], lines 1-7.

Claim 20 depends on claim 16 and adds the limitation that the information command layer script is further defined as logic to manipulate the requested data, *see, e.g.*, Application at 18, ¶ [0062], lines 1-2; Application at 25, ¶ [0069], lines 1-2; Application at 26, ¶ [0070], lines 1-3; Application at 26, ¶ [0071], lines 1-2.

Claim 21 depends on claim 16 and adds the limitation that the information command layer script is processed automatically upon receiving the at least portion of the requested data from the information service layer, *see, e.g.*, Application at 26, ¶ [0070], lines 1-8.

Claim 22 depends on claim 16 and adds the limitation that the information command layer script is processed prior to receiving the formatted portion of the requested data from the information service layer *see, e.g.*, *id.*

Claim 23 depends on claim 16 and adds the limitation that processing the information command layer script to format the data includes formatting the data to extensible mark-up language reports, *see, e.g.*, Application at 3, ¶ [0010], lines 7-8; Application at 24, ¶ [0064], lines 2-4; Application at 25, ¶ [0069], lines 2-3; Application at 26, ¶ [0070], lines 1-3; Application at 26, ¶ [0071], lines 6-7.

Claim 24 depends on claim 16 and adds the limitation that processing the information command layer script to format the data includes formatting the data to a tab delimited file Application at 3, ¶ [0010], lines 7-8; Application at 24, ¶ [0064], lines 4-6; Application at 25, ¶ [0069], lines 2-3; Application at 26, ¶ [0070], lines 1-3.

Claim 25 depends on claim 16 and adds the limitation that processing the information command layer script to format the data includes formatting the data to extensible mark-up language tags, *see, e.g.*, Application at 3, ¶ [0010], lines 7-8; Application at 12, ¶ [0045], lines 2-4, Application at 24, ¶ [0064], lines 2-4; Application at 25, ¶ [0069], lines 2-3; Application at 26, ¶ [0070], lines 1-3; Application at 26, ¶ [0071], lines 8-9.

Claim 26 depends on claim 16 and adds the limitation that building the request for data includes generating an extensible mark-up language tag, *see, e.g.*, Application at 12, ¶ [0045], lines 8-11; Application at 13, ¶ [0045], lines 1-4.

Claim 27 depends on claim 16 and adds the limitation that the information service layer is executed on a first computer system, the method further comprising passing formatted data to an application on a second computer system, *see, e.g.*, Application at 16, ¶ [0056], lines 1-4;

Application at 24, ¶ [0066], line 4; Application at 25, ¶ [0069], lines 4-5; Application at 33, ¶ [0094], lines 1-5; Figure 5; Figure 8.

Claim 28 is independent and is directed to a system for processing data stored on a computer readable medium retrieved from an information service layer, *see, e.g.*, Application at 25, ¶ [0067], line 1; Application at 25, ¶ [0069], lines 1-2, the system comprising: an application programmable interface that receives a request for a data from an application, *see, e.g.*, Application at 6, ¶ [0026], lines 1-2; Application at 25, ¶ [0069], lines 5-7; a data requestor that receives the request for data from and a first script from the application programmable interface, *see, e.g.*, Application at 25, ¶ [0066], lines 2-5; Application at 25, ¶ [0069], lines 5-7; an information service layer that receives the first script from the data requestor, retrieves the data, and creates a response document based upon formatting described in the first script, *see, e.g.*, Application at 25, ¶ [0066], lines 2-7; Application at 25, ¶ [0067], lines 1-3; a data receiver that receives the response document from the information service layer, *see, e.g.*, Application at 25, ¶ [0067], line 1; an analyzer that receives the response document from the data receiver and the analyzer analyzes the response document from the data receiver based on logic or instructions embedded in a second script received from the application programmable interface, *see, e.g.*, Application at 25, ¶ [0069], lines 1-2; Application at 25, ¶ [0069], lines 5-7; Application at 26, ¶ [0070], lines 1-3, and wherein the analyzer executes the logic or instructions embedded in the second script on the response document to analyze a metric associated with the application, *see, e.g.*, Application at 26, ¶ [0071], lines 1-7, and the analyzer creates an output document, *see, e.g.*, Application at 25, ¶ [0069], lines 1-5; Application at 26, ¶ [0071], lines 1-7; a writer that receives the output document and formats the output document into an output formatted document, *see, e.g.*, Application at 25, ¶ [0069], lines 2-5, and wherein the output formatted document is transmitted to the application programmable interface, *see, e.g.*, Application at 25, ¶ [0069], lines 4-7, and wherein the output formatted document is transmitted from the application programmable interface to the application, *see, e.g.*, *id.*

Claim 29 depends on claim 28 and adds the limitation that the first script comprises an information service layer script, *see, e.g.*, Application at 25, ¶ [0066], lines 5-8; Application at 26, ¶ [0070], lines 9-11.

Claim 30 depends on claim 28 and adds the limitation that the second script comprises an information command layer script, *see, e.g.*, Application at 26, ¶ [0070], lines 1-3.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 1, 6-8, 11, 16, 18, and 20-30 are unpatentable under 35 U.S.C. § 103(a) as being obvious over Draper et al. (U.S. Patent Number 7,152,062) (hereinafter *Draper*) in view of Brown et al. (U.S. Patent Publication 2006/0265260) (hereinafter *Brown*).
2. Whether claims 4, 5, and 19 are unpatentable under 35 USC § 103(a) as being obvious over Draper in view of Brown and in further view of Zarb (U.S. Patent Publication 2004/0039619) (hereinafter *Zarb*).

VII. ARGUMENT

The Examiner has failed to establish a *prima facie* case of obviousness with respect to claims 1-20. As noted by the United States Supreme Court in *Graham v. John Deere Co. of Kansas City*, an obviousness determination begins with a finding that **"the prior art as a whole in one form or another contains all" of the elements of the claimed invention**. *See Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 22 (U.S. 1966). Appellants respectfully submit that *Draper* and *Brown*, alone or in combination, do not contain all of the elements of claims 1, 4-8, 11, 16, and 18-30. Further, *Zarb* does not cure the deficiencies of *Draper* in view of *Brown*.

Draper is directed to techniques for encapsulating a query definition. S*ee Draper* at Abstract. In particular, *Draper* teaches an XML document representing a query definition including a query specification, a data source identifier, and optionally, results transform. *See id.* While *Draper* may disclose transforming the results of the query, *Draper* does not disclose performing calculations on the results of the query. Rather, *Draper* merely discloses that transforming results is transforming the results to a canonical format. *See Draper* at column 6, lines 4-8. Further, *Draper* merely discloses that formatting is performed once (i.e., the results are merely formatted once into canonical format when the results are received). *See id. Draper* does not disclose formatting requested data twice.

Brown is directed to a system and method that permits a user to browse and compare products of a merchant using information about a competitor. *See Brown* at Abstract. While *Brown* discloses a script processor 204 that performs a calculation on retrieved data, *see Brown* at 8, ¶ [0098], *Brown* does not disclose that the script processor 204 performs calculations according to an information command layer script received from an application program interface. In fact, a text search of *Brown* for "application program interface" and "API" produced no results. Furthermore, *Brown* does not teach or suggest that the data retrieved by the script processor 204 from the tables has been retrieved using an information service layer script and formatted based upon the formatting described in the information service layer script script.

Brown discloses that the script processor 204 executes on the merchant computer 102 (i.e., server side) and serves a plurality of customer browsers 214. S*ee Brown* at 3, ¶ [0035]; *Brown* at 8, ¶ [0098]; *Brown* at Fig. 2. In contrast, *Draper* discloses that the lenses, which perform functions such as transforming the results, execute on the client side. *See Draper* at Fig. 2. One of ordinary skill in the art would not have been motivated to combine the teachings of *Draper* and *Brown* and modify *Brown* to move the script processor 204 to the client side (e.g., customer browsers 214) because doing so would require each customer browser 214 to have a script processor 204, which would destroy the teachings of *Brown* (i.e., the script processor serving a plurality of customer browsers 214) and greatly increase costs to the merchant. Moreover, for the sake of argument, if one of ordinary skill in the art were motivated to combine the teachings of *Draper* and *Brown*, the combination would result in the script processor 204 performing calculations (i.e., performing the alleged information command layer script) on the server side prior to the lenses performing their functions (i.e., performing the alleged information service layer script) on the client side. *See Brown* at Fig. 2; *Draper* at Fig. 2; *Draper* at column 6, lines 4-8. Performing the information command layer script prior to the information service layer script is contrary to the limitations of claim 1.

Accordingly, *Draper* in view of *Brown*, alone or in combination, do not disclose an analyzer that analyzes the at least a portion of the requested data wherein the analyzer performs calculations on the at least a portion of the requested data according to the at least one information command layer script received from the application program interface. Further, *Draper* in view of *Brown*, alone or in combination, do not disclose formatting the requested data twice. Appellants respectfully submit that *Zarb* does not cure the deficiencies of *Draper* and *Brown* noted above.

These distinctions will be discussed in greater detail in the analysis of the pending claims that follows.

A. 35 U.S.C. § 103 Rejections - Claims 1, 6-8, 11, 16, 18, and 20-30.

1. Whether claims 1, 6-8, 11, 16, 18, and 20-30 are unpatentable under 35 U.S.C. § 103(a) as being obvious over *Draper* in view of *Brown*.

a. Claims 1, 6-8, 11, 16, 18, and 20-30 were wrongly rejected because neither *Draper* nor *Brown*, alone or in combination, disclose an analyzer that analyzes the at least a portion of the requested data wherein the analyzer performs calculations on the at least a portion of the requested data according to the at least one information command layer script received from the application program interface.

Claim 1 recites "an analyzer that receives the response from the receiver component, and the analyzer analyzes the at least a portion of the requested data wherein the analyzer performs calculations on the at least a portion of the requested data according to the at least one information command layer script received from the application program interface."

Pages 3-4 of the Final Office Action stated that "Draper does not expressly teach an analyzer that receives the response from the receiver component, and the analyzer analyzes the at least a portion of the requested data where in the analyzer performs calculations on the at least a portion of the requested data according to the at least some information command layer script received from the application program interface." Thus, the Final Office Action relied on *Brown* to teach the above cited limitation of claim 1. More specifically, page 4 of the Final Office Action stated that the claimed analyzer is taught by the "script processor 204" of *Brown* as disclosed in paragraphs 0097-0099 of *Brown*. Appellants respectfully disagree.

Appellants note that claim 1 requires that "the analyzer performs calculations on the…requested data according to the…information command layer script *received from the application program interface*" (emphasis added). Appellants respectfully submit that the "script processor 204" of *Brown* cannot be interpreted as the claimed analyzer because *Brown* does not teach or suggest that the script processor 204 performs calculations according to an information command layer script (hereinafter "ICL script") received *from an application program interface*. In fact, a text search of *Brown* for "application program interface" and "API" produced no results.

Additionally, Appellants note that claim 1 requires a "response formatted based upon the formatting described in the information service layer script, and the response including at least a portion of the requested data" and "an analyzer that receives the response from the receiver component, and…performs calculations on the at least a portion of the requested data." While paragraphs 0097 and 0098 of *Brown* may disclose that the script processor 204 issues queries to retrieve data from tables and performs calculations on the retrieved data, Appellants respectfully submit that the "script processor 204" of *Brown* cannot be interpreted as the claimed analyzer because *Brown* does not teach or suggest that the data retrieved by the script processor 204 from the tables has been retrieved using an information service layer script (hereinafter "ISL script") and formatted based upon the *formatting described in the ISL script*.

Furthermore, Appellants respectfully submit that it would not have been obvious to combine the teachings of *Draper* with the teachings of *Brown*. Referring to Fig. 2 of *Brown*, Appellants note that the script processor 204, which is being interpreted as the claimed analyzer, is located on a server which is separate and distinct from the customer browsers 214 (e.g., clients). In contrast, Fig. 2 of *Draper* discloses that the lenses and lens API are on the client side (i.e., elements 210-a and 210-b). Pages 4-5 of the Final Office Action stated, "A person having ordinary skill in the art would have been motivated to modify the teachings of Draper with the teachings of Brown by configuring Draper's invention with an information command layer script (i.e., script executed by script processor in accordance with ¶0097-¶0098, of Brown) to perform a calculation on the resulting data retrieved from the execution of one or more of Draper's lenses (see col. 4, lines 5-col. 5, line 24, read as the information service script)." Thus, the Final Office Action suggests that one of ordinary skill in the art would have been motivated to take the script processor 204 of *Brown*, which *Brown* discloses is located on the server side, and move it to the client side of *Draper* where the lenses are located.

Appellants respectfully submit that one of ordinary skill in the art would not have been motivated to make such a modification of *Brown*. *Brown* discloses in paragraph 0035 that the "script processor 204 runs on the merchant computer 102." Referring to Fig. 2 of *Brown*, the script processor 204 on the merchant computer 102 serves a plurality customer browsers 214. Moving the script processor 204 to the client side, as suggested by the Final Office Action, would require that each merchant provide each customer browser 214 with a script processor 204. Such a modification would destroy the teachings of Fig. 2 of *Brown* because each script processor 204 would merely serve a single customer browser 214, and not a plurality of customer browsers 214. Furthermore, having each merchant provide multiple script processors 204 for each customer browser 214 would greatly increase the costs for each merchant. For example, the merchant would have to purchase multiple licenses and pay multiple license fees, as opposed to paying a single licensing fee if single script processor 204 executed on the merchant's computer as *Brown* teaches.

Moreover, assuming for the sake of argument that one of ordinary skill were motivated to modify the teachings of *Draper* with the teachings of *Brown*, the combination would not result in "perform[ing] a calculation on the resulting data retrieved from the execution of one or more Draper's lenses," as suggested by the Final Office Action. Rather, because the script processor 204 of *Brown* is on the server side, referring to Fig. 2 of *Draper*, the combination of *Draper* in view of *Brown* would result in the script processor performing calculations on the server side (i.e., elements 220-a and 220-b) before the lenses receive any data. If *Draper* were modified in such a way so as to include the teachings of the script processor 204 of *Brown*, the alleged ICL script performed by the script processor 204 on the server side of *Draper* (i.e., elements 220-a and 220-b) would be performed prior to the functions performed by the lenses of *Draper*, which are alleged to be the claimed ISL script. Performing the ICL script prior to the ISL script is contrary to the limitations of claim 1. Accordingly, Appellants respectfully submit that neither *Draper* nor *Brown*, alone or in combination, disclose an analyzer that analyzes the at least a portion of the requested data wherein the analyzer performs calculations on the at least a portion of the requested data according to the at least one information command layer script received from the application program interface.

b. Claims 1, 6-8, 11, 16, 18, and 20-30 were wrongly rejected because neither *Draper* nor *Brown* disclose formatting the requested data twice.

Claim 1 recites "a receiver component communicating with the information service layer that receives a response from the information service layer, the response *formatted* based upon the formatting described in the information service layer script, and the response including at least a portion of the requested data;...and a writer component *formatting* the analyzed portion of the response based upon a desired format" (emphasis added). Thus, claim 1 requires that the requested data is formatted twice: (1) the response formatted based upon the formatting described in the ISL script and (2) the writer component formats the analyzed portion of the response.

Page 3 of the Final Office Action relied on column 3, lines 27-44 and col. 6, lines 2-9 of *Draper* to disclose the receiver component and column 6, lines 5-10 of *Draper* to teach the writer component. Appellants respectfully submit that column 3, lines 2-9 of *Draper* is irrelevant to the claimed feature because it merely discloses a reference for defining a query language to support XML, which has nothing to do with formatting based on information in an ISL script.

Column 6, lines 27-44 of *Draper* merely discloses receiving the results of the query from the server, and then transforming results of the query into a canonical format using the API. Thus, Appellants respectfully submit that the received response in *Draper* is not already formatted, but rather is formatted after receipt by the function of the API on the client. In contrast, the plain meaning of claim 1 is that the receiver component receives a response that is already formatted based on the formatting described in the ISL script.

Column 6, lines 5-10 of *Draper* discloses that "the function of the API that initiated the execution of the query may then use the results transform of the lens to transform the results to a canonical format before returning the results to the application program." Appellants note that column 6, lines 5-10 of *Draper*, which the Final Office Action alleges teaches the claimed writer component, merely discloses the same thing (i.e., transforming results to canonical format) as column 6, lines 27-44 of *Draper*, which is discussed above and relied on by the Final Office Action to teach the receiver component. Thus, Appellants respectfully submit that *Draper* merely discloses performing a single formatting once (i.e., transforming the results to canonical format after the results are received). In contrast, claim 1 requires that requested data is formatted twice. More specifically, claim 1 requires that the writer component format data that has already been formatted once before based on the ISL script. *Draper* does not disclose that the results that are transformed by the lenses to canonical format were formatted once before, let alone formatted once before based on an ISL script. Accordingly, Appellants respectfully submit that neither *Draper* nor *Brown*, alone or in combination, disclose formatting the requested data twice.

For at least the reasons established above in sections VII.A.1.a. and VII.A.1.b., Appellants respectfully submit that independent claim 1 is not taught or suggested by *Draper* in view of *Brown*. Dependent claims 6-8 and 11 depend directly or indirectly from independent claim 1 and incorporate all of the limitations thereof. Accordingly, for at least the reasons established above in sections VII.A.1.a. and VII.A.1.b., Appellants respectfully submit that claims 6-8 and 11 are not taught or suggested by *Draper* in view of *Brown*.

Independent claim 16 includes limitations substantially similar to the limitations discussed in sections VII.A.1.a. and VII.A.1.b. above. For example, claim 16 recites in part,

> [R]eceiving at least a portion of the requested data from the information service layer, wherein the at least a portion of the requested data is formatted according to the information service layer script; and processing the at least a portion of the requested data received from the information service layer according to an information command layer script, wherein the information command layer script comprises tagged data, wherein the processing comprises: parsing the information command layer script; interpreting embedded instructions from the information command layer script; performing a calculation on the at least a portion of the requested data based upon the embedded instructions to analyze a metric associated with the application; and formatting the analyzed portion of the requested data based on a desired format.

Therefore, the arguments of sections VII.A.1.a. and VII.A.1.b. are hereby repeated for independent claim 15. Accordingly, for at least the reasons established above in sections

VII.A.1.a. and VII.A.1.b., Appellants respectfully submit that claim 16 is not taught or suggested by *Draper* in view of *Brown*.

Dependent claims 18 and 20-27 depend directly or indirectly from independent claim 16 and incorporate all of the limitations thereof. Therefore, the arguments of sections VII.A.1.a. and VII.A.1.b are hereby repeated for claims 18 and 20-27. Accordingly, for at least the reasons established above in sections VII.A.1.a. and VII.A.1.b, Appellants respectfully submit that claims 16 and 18-20 are not taught or suggested by *Draper* in view of *Brown*.

Independent claim 28 includes limitations substantially similar to the limitations discussed in sections VII.A.1.a. and VII.A.1.b. above. For example, claim 28 recites in part,

> [A]n information service layer that receives the first script from the data requestor, retrieves the data, and creates a response document based upon formatting described in the first script; a data receiver that receives the response document from the information service layer; an analyzer that receives the response document from the data receiver and the analyzer analyzes the response document from the data receiver based on logic or instructions embedded in a second script received from the application programmable interface, and wherein the analyzer executes the logic or instructions embedded in the second script on the response document to analyze a metric associated with the application, and the analyzer creates an output document; a writer that receives the output document and formats the output document into an output formatted document.

Therefore, the arguments of sections VII.A.1.a. and VII.A.1.b. are hereby repeated for independent claim 28. Accordingly, for at least the reasons established above in sections VII.A.1.a. and VII.A.1.b., Appellants respectfully submit that claim 28 is not taught or suggested by *Draper* in view of *Brown*.

Dependent claims 29 and 30 depend directly or indirectly from independent claim 28 and incorporate all of the limitations thereof. Therefore, the arguments of sections VII.A.1.a. and VII.A.1.b are hereby repeated for claims 29 and 30. Accordingly, for at least the reasons established above in sections VII.A.1.a. and VII.A.1.b, Appellants respectfully submit that claims 29 and 30 are not taught or suggested by *Draper* in view of *Brown*.

2. Whether claims 4, 5, and 19 are unpatentable under 35 USC § 103(a) as being obvious over *Draper* in view of *Brown* and in further view of *Zarb*.

a. Claims 4 and 5 depend from claim 1 and were wrongly rejected for at least the reasons established in sections VII.A.1.a. and VII.A.1.b.

Dependent claims 4 and 5 depend directly or indirectly from independent claim 1 and incorporate all of the limitations thereof. Therefore, the arguments of sections VII.A.1.a. and VII.A.1.b. are hereby repeated for claims 4 and 5. Further, *Zarb* is directed to a system and a method for analyzing an organization. *See, e.g.*, *Zarb* at Abstract. The Final Office Action relied on *Zarb* in the rejection of claims 4 and 5 for disclosure regarding collecting and monitoring data associated with invoicing cycles. Appellants submit that none of the disclosure relied on in *Zarb* cures the deficiencies of *Draper* and *Brown* noted above. Accordingly, for at least the reasons established above in sections VII.A.1.a. and VII.A.1.b., Appellants respectfully submit that claims 4 and 5 are not taught or suggested by *Draper* in view of *Brown* and *Zarb*.

b. Claim 19 depends from claim 16 and was wrongly rejected for at least the reasons established above in sections VII.A.1.a. and VII.A.1.b.

Dependent claim 19 depends directly or indirectly from independent claim 16 and incorporates all of the limitations thereof. Therefore, the arguments of sections VII.A.1.a. and VII.A.1.b. are hereby repeated for claim 19. Further, as discussed above in section VII.A.2.a., *Zarb* is directed to a system and a method for analyzing an organization. *See, e.g.*, *Zarb* at Abstract. The Final Office Action relied on Zarb in the rejection of claim 19 for disclosure regarding collecting and monitoring data associated with invoicing cycles. Appellants submit that none of the disclosure relied on in *Zarb* cures the deficiencies of *Draper* and *Brown* noted above. Accordingly, for at least the reasons established above in sections VII.A.1.a. and VII.A.1.b., Appellants respectfully submit that claim 19 is not taught or suggested by *Draper* in view of *Brown* and *Zarb*.

VIII. CONCLUSION

In view of the above arguments the Appellants respectfully request that the Final Rejection of the claims be reversed and the case advanced to issue. Should the Examiner feel that a telephone interview would advance prosecution of the present application, the Appellants invite the Examiner to call the attorneys of record.

The Commissioner is hereby authorized to charge payment of any further fees associated with any of the foregoing papers submitted herewith, or to credit any overpayment thereof, to Deposit Account No. 21-0765, of Sprint Communications Company L.P.

Respectfully submitted,
CONLEY ROSE, P.C.

Date: January 25, 2010

/Michael W. Piper/
Michael W. Piper
Reg. No. 39,800

ATTORNEY FOR APPELLANTS

CONLEY ROSE, P.C.
5601 Granite Parkway, Suite 750
Plano, Texas 75024
(972) 731-2288
(972) 731-2289 (facsimile)

IX. CLAIMS APPENDIX

1. A system for processing data stored on a computer readable medium retrieved from an information service layer, the system comprising:

an application program interface that receives a request for data from an application, and that provides at least one information service layer script and at least one information command layer script;

a requestor component that builds a request for data using the at least one information service layer script, and transmits the built request for data to the information service layer;

a receiver component communicating with the information service layer that receives a response from the information service layer, the response formatted based upon the formatting described in the information service layer script, and the response including at least a portion of the requested data;

an analyzer that receives the response from the receiver component, and the analyzer analyzes the at least a portion of the requested data wherein the analyzer performs calculations on the at least a portion of the requested data according to the at least one information command layer script received from the application program interface, wherein instructions are embedded within the at least one information command layer script using a tag-based markup language, wherein the analyzer parses the information command layer script to obtain the instructions embedded in the information command layer script to analyze~~s~~ a metric associated with the application based on the at least a portion of the requested data, and the analyzer creates an analyzed portion of the response; and a writer component formatting the analyzed portion of the response based upon a desired format,

wherein the application program interface sends the formatted analyzed portion of the response to the application.

2. - 3. (Canceled)

4. The system of Claim 1, wherein the analyzer further verifies cycles associated with the application.

5. The system of Claim 4, wherein the cycles associated with the application is further defined as invoicing cycles.

6. The system of Claim 1, wherein the writer component formats the analyzed portion of the response as an extensible mark-up language report.

7. The system of Claim 1, wherein the writer component formats the analyzed portion of the response as an extensible mark-up language tag.

8. The system of Claim 1, wherein the writer component formats the analyzed portion of the response as character delimited.

9. -10. (Canceled)

11. The system of Claim 1, wherein the application program interface promotes communications between at least one application resident on at least one other computer system.

12. - 15. (Canceled)

16. A method for processing data retrieved from an information service layer; the method comprising:

building a request for data, wherein the request comprises an information service layer script provided by an application program interface;

submitting the request for data to an information service layer;

receiving at least a portion of the requested data from the information service layer, wherein the at least a portion of the requested data is formatted according to the information service layer script; and

processing the at least a portion of the requested data received from the information service layer according to an information command layer script, wherein the information command layer script comprises tagged data, wherein the processing comprises:

parsing the information command layer script;

interpreting embedded instructions from the information command layer script;

performing a calculation on the at least a portion of the requested data based upon the embedded instructions to analyze a metric associated with the application; and

formatting the analyzed portion of the requested data based on a desired format;

wherein the information command layer script further comprises instructions to monitor a process based on the requested data and spawn an event based upon a trigger related to the requested data.

17. (Canceled)

18. The method of Claim 16, wherein the process is defined as an application operation.

19. The method of Claim 16, wherein the process is defined as an invoicing cycle.

20. The method of Claim 16, wherein the information command layer script is further defined as logic to manipulate the requested data.

21. The method of Claim 16, wherein the information command layer script is processed automatically upon receiving the at least portion of the requested data from the information service layer.

22. The method of Claim 16, wherein the information command layer script is processed prior to receiving the formatted portion of the requested data from the information service layer.

23. The method of Claim 16, wherein processing the information command layer script to format the data includes formatting the data to extensible mark-up language reports.

24. The method of Claim 16, wherein processing the information command layer script to format the data includes formatting the data to a tab delimited file.

25. The method of Claim 16, wherein processing the information command layer script to format the data includes formatting the data to extensible mark-up language tags.

26. The method of Claim 16, wherein building the request for data includes generating an extensible mark-up language tag.

27. The method of Claim 16, wherein the information service layer is executed on a first computer system, the method further comprising passing formatted data to an application on a second computer system.

28. A system for processing data stored on a computer readable medium retrieved from an information service layer, the system comprising:

an application programmable interface that receives a request for a data from an application;

a data requestor that receives the request for data from and a first script from the application programmable interface;

an information service layer that receives the first script from the data requestor, retrieves the data, and creates a response document based upon formatting described in the first script;

a data receiver that receives the response document from the information service layer;

an analyzer that receives the response document from the data receiver and the analyzer analyzes the response document from the data receiver based on logic or instructions embedded in a second script received from the application programmable interface, and wherein the analyzer executes the logic or instructions embedded in the second script on the response document to analyze a metric associated with the application, and the analyzer creates an output document;

a writer that receives the output document and formats the output document into an output formatted document, and wherein the output formatted document is transmitted to the application programmable interface, and wherein the output formatted document is transmitted from the application programmable interface to the application.

29. The system of Claim 28, wherein the first script comprises an information service layer script.

30. The system of Claim 28, wherein the second script comprises an information command layer script.

X. EVIDENCE APPENDIX

None.

XI. RELATED PROCEEDINGS APPENDIX

None.